

AMENDMENTS TO THE CLAIMS

Applicant submits below a complete listing of the current claims, including marked-up claims with insertions indicated by underlining and deletions indicated by strikeouts and/or double bracketing. This listing of claims will replace all prior versions, and listings, of claims in the application:

1-95. (Canceled).

96. (Currently Amended) A method of designing a software system, comprising:
defining a set of functional requirements that describe what the software system is to achieve;

defining a set of design parameters, where each design parameter in the set satisfies at least one of the functional requirements;

decomposing the set of functional requirements and design parameters to create a hierarchy of functional requirements and a hierarchy of design parameters, wherein ~~at least one functional requirement of the set of~~ the hierarchy of functional requirements ~~[[is a]]~~ comprises at least one parent functional requirement at a ~~[[first]]~~ top level in the hierarchy of functional requirements and is decomposed into at least two ~~[[child]]~~ leaf functional requirements at a ~~second bottom~~ level in the hierarchy of functional requirements that is below the first level, ~~[[and]]~~ wherein the at least two ~~[[child]]~~ leaf functional requirements collectively accomplish the at least one parent functional requirement, and wherein the hierarchy of design parameters comprises at least one parent design parameter at a top level in the hierarchy of design parameters and is decomposed into at least two leaf design parameters at a bottom level in the hierarchy of design parameters, wherein the at least two leaf functional requirements collectively describe the at least one parent design parameter;

defining a design matrix that comprises a plurality of rows, a plurality of columns, and a plurality of cells, each of which is formed by the intersection of one of the plurality of rows and one of the plurality of columns, wherein each of the plurality of rows corresponds to at least one of the leaf functional requirements, each column of the design matrix corresponds to at least one of the leaf design parameters, and each cell corresponds to the intersection of the leaf functional

requirements and one of the leaf design parameters, and wherein a cell in the design matrix is marked with an indicator when its corresponding leaf design parameter satisfies its corresponding leaf functional requirement that maps each design parameter in the hierarchy of design parameters to the at least one functional requirement in the hierarchy of functional requirements that the respective design parameter satisfies; and

using the design matrix to define an object-oriented structure of the software system by performing acts of; ~~wherein at least one functional requirement in the hierarchy of functional requirements represents a software object of the software system, and wherein at least one design parameter in the hierarchy of design parameters represents an attribute of the software object.~~

a) creating an object-oriented class that corresponds to a parent design parameter in the hierarchy of design parameters;

b) identifying attributes of the object-oriented class by representing the leaf design parameters of the parent design parameter as the attributes of the object-oriented class;

c) for each of the leaf design parameters identified in the act b), determining from the design matrix the leaf functional requirement that satisfies the design parameter; and

d) representing each leaf functional requirement determined in the act c) as a method of the object-oriented class, wherein the method for each leaf functional requirement is a method to perform the task defined by the leaf functional requirement.

97. (Canceled)

98. (Previously Presented) The method of claim 96, wherein the act of defining the set of design parameters further comprises determining the set of design parameters by mapping the set of functional requirements into a physical implementation domain.

99. (Previously Presented) The method of claim 96, further comprising an act of determining if the design matrix is decoupled.

100. (Previously Presented) The method of claim 99, further comprising an act of, when it is determined that the design matrix is not decoupled, manipulating the design matrix into lower triangular form.

101-104. (Canceled)

105. (Currently Amended) At least one computer readable medium encoded with instructions that, when executed on a computer system, perform a method of allowing a user to define a software system, the method comprising:

allowing the user to define a set of functional requirements that describe what the software system is to achieve;

allowing the user to define a set of design parameters, where each design parameter in the set satisfies at least one of the functional requirements;

allowing the user to decompose the set of functional requirements and design parameters to create a hierarchy of functional requirements and a hierarchy of design parameters, wherein at ~~least one functional requirement of the set of~~ the hierarchy of functional requirements ~~[[is a]]~~ comprises at least one parent functional requirement at a ~~[[first]]~~ top level in the hierarchy of functional requirements and is capable of being decomposed into at least two ~~[[child]]~~ leaf functional requirements at a ~~second~~ bottom level in the hierarchy of functional requirements ~~that is below the first level,~~ [[and]] wherein the at least two ~~[[child]]~~ leaf functional requirements collectively accomplish the at least one parent functional requirement, and wherein the hierarchy of design parameters comprises at least one parent design parameter at a top level in the hierarchy of design parameters and is decomposed into at least two leaf design parameters at a bottom level in the hierarchy of design parameters, wherein the at least two leaf functional requirements collectively describe the at least one parent design parameter;

allowing the user to define a design matrix that comprises a plurality of rows, a plurality of columns, and a plurality of cells, each of which is formed by the intersection of one of the plurality of rows and one of the plurality of columns, wherein each of the plurality of rows corresponds to at least one of the leaf functional requirements, each column of the design matrix

corresponds to at least one of the leaf design parameters, and each cell corresponds to the intersection of the leaf functional requirements and one of the leaf design parameters, and wherein a cell in the design matrix is marked with an indicator when its corresponding leaf design parameter satisfies its corresponding leaf functional requirement that maps each design parameter in the hierarchy of design parameters to the at least one functional requirement in the hierarchy of functional requirements that the respective design parameter satisfies; and

using the design matrix to define an object-oriented structure of the software system by performing acts of; wherein at least one functional requirement in the hierarchy of functional requirements represents a software object of the software system, and wherein at least one design parameter in the hierarchy of design parameters represents an attribute of the software object.

a) creating an object-oriented class that corresponds to a parent design parameter in the hierarchy of design parameters;

b) identifying attributes of the object-oriented class by representing the leaf design parameters of the parent design parameter as the attributes of the object-oriented class;

c) for each of the leaf design parameters identified in the act b), determining from the design matrix the leaf functional requirement that satisfies the design parameter; and

d) representing each leaf functional requirement determined in the act c) as a method of the object-oriented class, wherein the method for each leaf functional requirement is a method to perform the task defined by the leaf functional requirement.

106. (Canceled).

107. (Previously Presented) The at least one computer readable medium of claim 105, wherein the act of allowing the user to define the set of design parameters further comprises allowing the user to map the set of functional requirements into a physical implementation domain to determine the set of design parameters.

108-113. (Canceled)